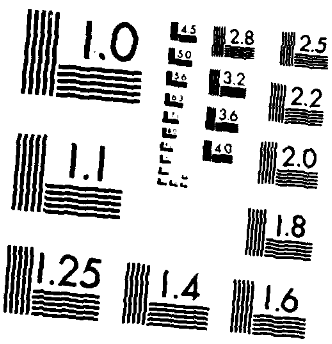
AD-A173 264 A PROGRAM FOR ESTIMATING UNCERTAINTIES IN QUANTILE ESTIMATES DERIVED FROM EMPIRICAL PEARSON FITS(U) STANFORD UNIV CA DEPT OF STATISTICS I JOHNSTONE 1/1

UNCLASSIFIED 14 OCT 86 TR-381 N00014-86-K-0156 F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

A PROGRAM FOR ESTIMATING UNCERTAINTIES IN QUANTILE ESTIMATES DERIVED FROM EMPIRICAL PEARSON FITS

BY

IAIN JOHNSTONE

TECHNICAL REPORT NO. 381
OCTOBER 14, 1986

PREPARED UNDER CONTRACT
N00014-86-K-0156 (NR-042-267)
FOR THE OFFICE OF NAVAL RESEARCH

DEPARTMENT OF STATISTICS
STANFORD UNIVERSITY
STANFORD, CALIFORNIA

A PROGRAM FOR ESTIMATING UNCERTAINTIES IN QUANTILE ESTIMATES DERIVED FROM EMPIRICAL PEARSON FITS

BY

IAIN JOHNSTONE

§1. Overview

The attached program assesses the variability in the estimate of an upper percentage point of a Pearson curve when its moments are estimated fr m data. This overview only aims to describe its use in broadest terms. Section 2 describes the theory and Section 3 the program in detail. A Fortran listing follows in an Appendix.

The program assumes that a sample of size *nsamp* has been drawn from a Pearson type IV distribution (Pearson type VII - the student *t* family - is also included, but the Gaussian law is not). On the basis of this sample it is assumed that the first four raw moments have been empirically estimated by the method of moments. It is desired to estimate an upper quantile . It is assumed that this is done by solving for the quantile of the Pearson curve with the same first four moments as the sample. Let us call the estimate obtained in this way $x0$. The purpose of the program is to numerically assess the uncertainty in $x0$ that is propagated from the uncertainty in the empirical moment estimates based on the sample of size *nsamp*. This calculation is done via the "delta method" and hence is based on a simple asymptotic Taylor series approximation that is discussed in the attached documentation.

The program is operated as follows : it contains a main program quantile.f and subroutine files subzero.f and subp4df.f. Suppose that the program is compiled into an executable program "quantile". On the Stanford Statistics department VAX750/UNIX4.2bsd, the program might be run as follows:

```
quantile <input >output
```

This specifies that the input data (discussed below) is read in from the data file "input" and the results written on an output file "output" The output has been formatted so that it can then be read into an appropriate statistical package (S, from Bell Labs, is what I had in mind) for further analysis.

Here is a toy example of an input file :

```
n
0  1  .2  3.9  100  .05
0  1  .2  3.9  100  .005
0  1  .2  3.9  100  .0005
0  1  .2  3.9  100  .00005
```

The first line contains a character indicating whether long output is desired (*y* for yes , *n* for no) : the long output of intermediate calculations, described later, that is mostly of interest during debugging. Subsequent lines contain six columns entered in free (ie Fortran list - directed) format:

Cols 1 - 4: μ, $\sigma 2$, $\beta 1$, $\beta 2$

= mean, var, skewness, kurtosis of generating distribution

5: nsamp = number of observations taken

6: eps = percentile that is to be estimated

Here is the output file produced from the above input :

long output?

mean	var	beta1	beta2	nsamp	eps	x0	dens	s.d.
0.	1.00	0.20	3.90	100.	0.500e-1	1.73	0.953e-1	0.189
0.	1.00	0.20	3.90	100.	0.500e-2	3.12	0.155e-1	0.416
0.	1.00	0.20	3.90	100.	0.500e-3	4.54	0.234e-2	0.661
0.	1.00	0.20	3.90	100.	0.500e-4	6.08	0.348e-3	0.775

The first line is a prompt which is useful only if input and output are occurring interactively at a terminal. The second line describes the contents of the columns : in addition to the input variables, we have

$x0$: the upper eps - th percentile corresponding to mean,var, beta1 and beta2

dens : the value of the Pearson curve density when evaluated at x0

s.d. : the approximate standard deviation of the estimate of $x0$ that is obtained by using the sample moments based on a sample of size *nsamp*

By selecting just the numerical output (or a subset of the columns) , further graphing of the output, statistical analysis, etc. is possible.

It should be remarked that the above program has NOT been fully debugged and checked . Obvious blunders have been removed and some tests that parameters lie within the Type IV/VII ranges and have sufficiently many moments have been built in. No claims are made for the quality of the code or algorithms. However the program should be useful for exploration of the uncertainty in[illegible]ed in empirical determination of threshholds.

Many authors have warned of the hazards involved in fitting Pearson curves on the basis of empirically determined moments. This program offers the capability for assessing (subject to the validity of the delta method approximation, of course !) quantitatively the size of these hazards.

As a further illustration of the output of the program, the following experiment was run. For

certain pairs of (β_1, β_2) falling in the Type IV (or VII) region

Case	1	2	3	4	5
β_1	0	0	1	1	2
β_2	3.3	4	5.25	6	7.5

Table 1.

and sample size $nsamp = 100$, the uncertainity in estimating $x0$ for $\epsilon = .05, .01, .005, .001, .0005, .0001, .00005, .00001$ is calculated. The two cases with zero skewness are scaled versions of the t distribution on 24 and 10 d.f. respectively.

The raw output is shown in Table 2. One interesting plot is to look at the coefficient of variation of the estimated percentile, namely s.d.$(x0/x0)$. These values are plotted against $-log_{10}(\epsilon)$ in the upper box of Figure 1, with the plotting character indicating the case number. Clearly as the skewness and kurtosis increase, so the coefficients of variation become increasingly unstable the further out one goes into the tail of the distribution. A second plot of the square roots of the coefficients of variation is shown for comparison.

Note that in the delta method calculations, the effect of increasing sample size $nsamp$ is simply to reduce the s.d.'s and coeff. of variation of $x_\epsilon(\beta)$ by $\sqrt{nsamp}$.

Figure 1. Coefficient of variation (and its square root) of $x_\varepsilon(\beta)$ for μ=0, σ^2=1, and 5 values of β specified in Table 1.

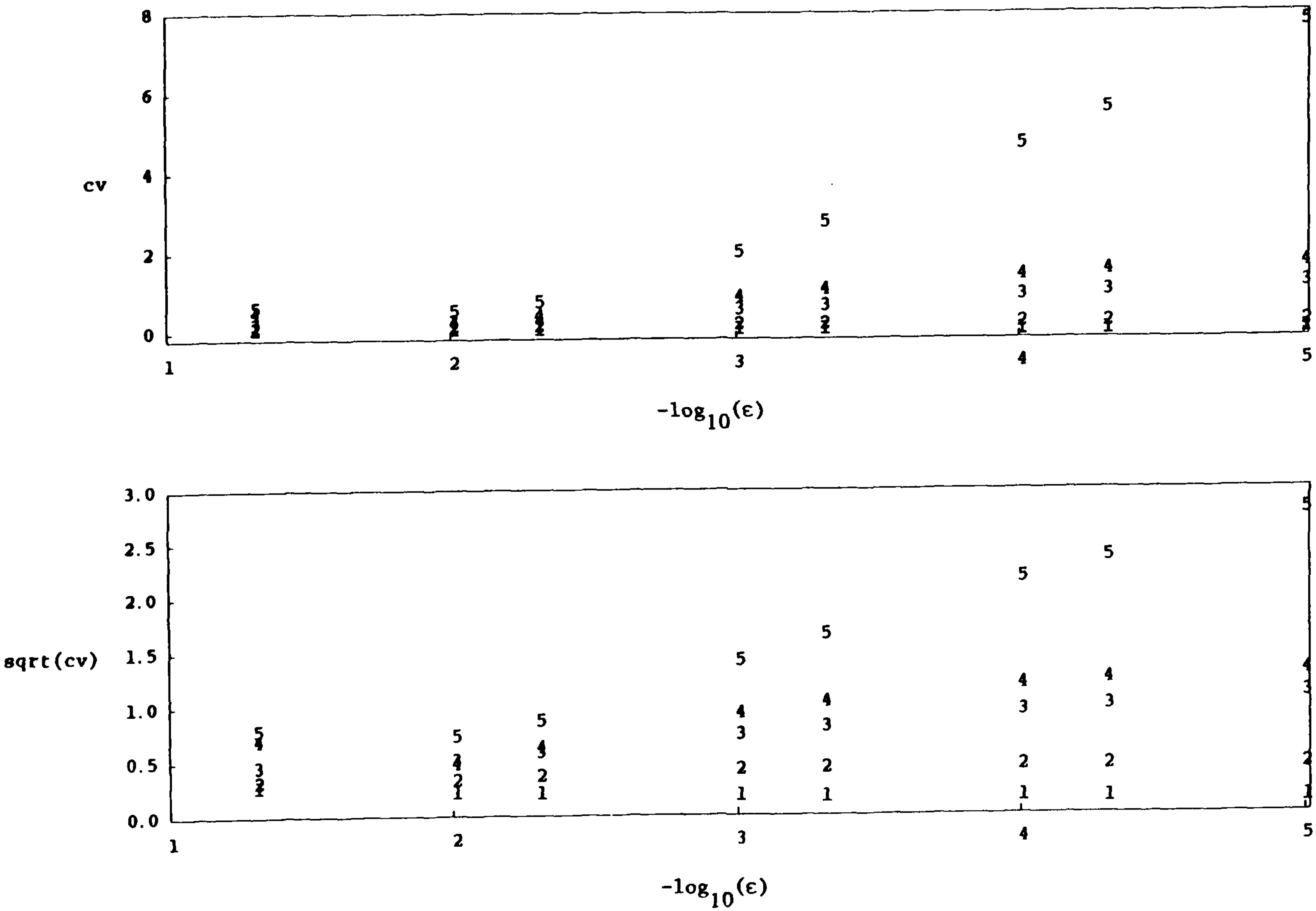


Table 2. Output from program for input parameters given in Table 1, together with values of nsamp and ε shown in columns 5 and 6.

mean	var	betal	beta2	nsamp	eps	x0	dens	s.d.
0.	1.00	0.	3.30	100.	0.500e-1	1.64	0.133	0.117
0.	1.00	0.	3.30	100.	0.100e-1	2.39	0.577e-1	0.980e-1
0.	1.00	0.	3.30	100.	0.500e-2	2.68	0.396e-1	0.964e-1
0.	1.00	0.	3.30	100.	0.100e-2	3.32	0.161e-1	0.949e-1
0.	1.00	0.	3.30	100.	0.500e-3	3.59	0.109e-1	0.934e-1
0.	1.00	0.	3.30	100.	0.100e-3	4.20	0.430e-2	0.875e-1
0.	1.00	0.	3.30	100.	0.500e-4	4.46	0.287e-2	0.843e-1
0.	1.00	0.	3.30	100.	0.100e-4	5.06	0.112e-2	0.764e-1
0.	1.00	0.	4.00	100.	0.500e-1	1.62	0.105	0.148
0.	1.00	0.	4.00	100.	0.100e-1	2.47	0.283e-1	0.257
0.	1.00	0.	4.00	100.	0.500e-2	2.83	0.156e-1	0.357
0.	1.00	0.	4.00	100.	0.100e-2	3.71	0.380e-2	0.613
0.	1.00	0.	4.00	100.	0.500e-3	4.10	0.205e-2	0.720
0.	1.00	0.	4.00	100.	0.100e-3	5.09	0.481e-3	0.947
0.	1.00	0.	4.00	100.	0.500e-4	5.56	0.256e-3	1.04
0.	1.00	0.	4.00	100.	0.100e-4	6.73	0.590e-4	1.24
0.	1.00	1.00	5.25	100.	0.500e-1	1.82	0.590e-1	0.352
0.	1.00	1.00	5.25	100.	0.100e-1	3.03	0.107e-1	0.779
0.	1.00	1.00	5.25	100.	0.500e-2	3.57	0.516e-2	1.17
0.	1.00	1.00	5.25	100.	0.100e-2	4.86	0.994e-3	2.62
0.	1.00	1.00	5.25	100.	0.500e-3	5.45	0.497e-3	3.48
0.	1.00	1.00	5.25	100.	0.100e-3	6.91	0.103e-3	6.01
0.	1.00	1.00	5.25	100.	0.500e-4	7.58	0.532e-4	7.33
0.	1.00	1.00	5.25	100.	0.100e-4	9.26	0.118e-4	10.9
0.	1.00	1.00	6.00	100.	0.500e-1	1.78	0.693e-1	0.827
0.	1.00	1.00	6.00	100.	0.100e-1	3.03	0.139e-1	0.675
0.	1.00	1.00	6.00	100.	0.500e-2	3.60	0.689e-2	1.36
0.	1.00	1.00	6.00	100.	0.100e-2	5.07	0.135e-2	4.35
0.	1.00	1.00	6.00	100.	0.500e-3	5.76	0.668e-3	6.04
0.	1.00	1.00	6.00	100.	0.100e-3	7.57	0.132e-3	10.5
0.	1.00	1.00	6.00	100.	0.500e-4	8.45	0.661e-4	12.7
0.	1.00	1.00	6.00	100.	0.100e-4	10.7	0.133e-4	17.9
0.	1.00	2.00	7.50	100.	0.500e-1	1.84	0.414e-1	1.11
0.	1.00	2.00	7.50	100.	0.100e-1	3.24	0.542e-2	1.67
0.	1.00	2.00	7.50	100.	0.500e-2	3.88	0.234e-2	2.83
0.	1.00	2.00	7.50	100.	0.100e-2	5.52	0.356e-3	10.9
0.	1.00	2.00	7.50	100.	0.500e-3	6.30	0.163e-3	17.1
0.	1.00	2.00	7.50	100.	0.100e-3	8.31	0.279e-4	38.8
0.	1.00	2.00	7.50	100.	0.500e-4	9.27	0.133e-4	51.7
0.	1.00	2.00	7.50	100.	0.100e-4	11.8	0.246e-5	91.2

§2. Sensitivity of Determination of Percentage Points of Pearson Curves when moments are estimated (Theoretical documentation to accompany program.)

Suppose that wish to estimate the percentage point x_ε, which for given ε solves the equation

(1) $$P(X > x) = \varepsilon.$$

Suppose that X is believed to follow a distribution within the Pearson family. Then the equation assumes the form

(2) $$F(\beta, x) = \varepsilon$$

where the vector β contains the parameters specifying the precise form of the distribution function F (for example the first four theoretical moments). If β is known, then equation (2) can be solved to yield the desired percentage point $x_\varepsilon(\beta)$. Such solutions have of course been tabulated for a limited selection of ε and β in the Pearson-Hartley *Biometrika* tables.

There is sometimes interest in situations in which ε is so small that the *Biometrika* tables are useless. The purpose of this discussion is to attempt a sensitivity analysis of the following types: how does an error in specification of β propagate into the resulting percentage point $x_\varepsilon(\beta)$? If β is estimated with a given standard deviation, what is the corresponding standard deviation of $x_\varepsilon(\beta)$? The analysis will be simple and approximate, based on the so-called "delta-method" (Rao (1973,§6g)).

To clarify the derivation, we regard ε as fixed and introduce the notation $h(\beta)$ for $x_\varepsilon(\beta)$. Thus the function h is implicitly defined by the equation

(3) $$F(\beta, h(\beta)) = \varepsilon.$$

The delta-method approximation simply says that if $\hat{\beta}$ is a $d \times 1$ random vector with mean $E\hat{\beta} = \beta$, then

(4) $$\mathrm{Var}_\beta h(\hat{\beta}) \doteq [Dh(\beta)]^t [\mathrm{Var}_\beta \hat{\beta}][Dh(\beta)]$$

Here Dh is a $d \times 1$ vector of partial derivatives, and $\mathrm{Var}_\beta \hat{\beta}$ is a $d \times d$ matrix. Now the implicit function theorem says (assuming necessary regularity conditions) that

(5) $$Dh(\beta) = -[D_2 F(\beta, h(\beta))]^{-1}[D_1 F(\beta, h(\beta))]$$

where D_1F and D_2F denote partial derivatives with respect to the first and second arguments of F, and are hence 1×1 and $d \times 1$ vectors respectively. Writing $f(\beta, x) = \frac{\partial}{\partial x} F(\beta, x)$ for the density corresponding to F, (5) reduces to

$$
(6) \qquad Dh(\beta) = \frac{-D_1F(\beta, h(\beta))}{f(\beta, h(\beta))},
$$

so that (4) becomes

$$
(7) \qquad \mathrm{Var}_\beta h(\beta) = \frac{1}{f^2}[D_1F^t][\mathrm{Var}_\beta \hat{\beta}][D_1F] \quad \text{evaluated at } (\beta, h(\beta)).
$$

In our example, the vector β of parameters describing the Pearson curve is taken to be $\beta = (\mu_1, \sigma^2, \beta_1, \beta_2)$. We assume that these will be estimated from i.i.d. data $X_1, \ldots, X_n$, which have been summarized through the first four raw moments $\hat{\mu}'_k = \frac{1}{n}\sum_{i=1}^n X_i^k$, $k = 1, 2, 3, 4$. (Central moments could equally well be used : raw moments are taken for simplicity of calculation here : the distinction will not affect the final result). Of couse, we have

$$
(8) \qquad \beta = \begin{pmatrix} \mu_1 \\ \sigma^2 \\ \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3^2/\mu_2^3 \\ \mu_4/\mu_2^2 \end{pmatrix} = \mathcal{G}\begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{pmatrix} = \mathcal{G}(\mu)
$$

so that if we estimate β by $\hat{\beta} = \mathcal{G}(\hat{\mu})$, then the delta method estimate of the variance of $\hat{\beta}$ is

$$
(9) \qquad \mathrm{Var}(\hat{\beta}) \doteq [D\mathcal{G}]\mathrm{Var}(\hat{\mu})[D\mathcal{G}]^t
$$

where

$$
(10) \qquad D\mathcal{G}_{ij} = \left(\frac{\partial \mathcal{G}_i}{\partial \mu_j}\right) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -3\mu_3^2/\mu_2^4 & 2\mu_3/\mu_2^3 & 0 \\ 0 & -2\mu_4/\mu_2^3 & 0 & 1/\mu_2^2 \end{pmatrix}
$$

It will be easier to work with the raw sample moments $\{\mu'_k\}$, which are related to the central sample moments by

$$
(11) \qquad \mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{pmatrix} = H(\mu') = \begin{pmatrix} \mu'_1 \\ \mu'_2 - \mu'^2_1 \\ \mu'_3 - 3\mu'_2\mu'_1, +2\mu'^3_1 \\ \mu'_4 - 4\mu'_3\mu'_1 + 6\mu'_2\mu'^2_1 - 3\mu'^4_1 \end{pmatrix}.
$$

Again applying the delta method, we arrive at

$$
(12) \qquad \mathrm{Var}(\hat{\beta}) \doteq [D\mathcal{G}][DH]\mathrm{Var}(\hat{\mu}')[DH]^t[D\mathcal{G}]^t,
$$

where

$$DH_{jk} = \left(\frac{\partial \mu_j}{\partial \mu'_k}\right) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -2\mu'_1 & 1 & 0 & 0 \\ -3\mu'_2 + 6\mu'^2_1 & -3\mu'_1 & 1 & 0 \\ -4\mu'_3 + 12\mu'_2\mu'_1 - 12\mu'^3_1 & 6\mu'^2_1 & -4\mu'_1 & 1 \end{pmatrix}. \tag{13}$$

Finally, we note that the entries in $\mathrm{Var}(\hat{\mu}')$ are obtained from

$$\begin{aligned} \mathrm{Cov}(\hat{\mu}'_k, \hat{\mu}'_{k'}) &= \frac{1}{n^2}\sum_i \sum_{i'} \mathrm{Cov}(X_i^k, X_{i'}^{k'}) \\ &= \frac{1}{n}\mathrm{Cov}(X_1^k, X_1^{k'}) = \frac{1}{n}[\mu'_{k+k'} - \mu'_k\mu'_{k'}]. \end{aligned} \tag{14}$$

Since $k \in \{1,2,3,4\}$, this would appear to require us to know the raw moments of the appropriate Pearson curve up through order 8.

§§2.1 Higher moments of the Pearson Type IV family:

Let $\bar{x}$ denote the 'independent variable' with origin referred to the mean of the distribution. Then the coefficients in the Pearson curve equation

$$\frac{1}{\bar{f}(\beta, \bar{x})}\frac{d\bar{f}}{d\bar{x}} = \frac{\bar{x} + \bar{a}}{\bar{b}_0 + \bar{b}_1\bar{x} + \bar{b}_2\bar{x}^2} \tag{15}$$

are related to $(\sigma^2, \beta_1, \beta_2)$ via the relations (compare eqn (3), p 39 of Elderton/Johnson (1969)):

$$\bar{a} = \sigma\sqrt{\beta_1}(\beta_2 + 3)/\Gamma \qquad \Gamma = 2(5\beta_2 - 6\beta_1 - 9)$$

$$\bar{b}_0 = -\sigma^2(4\beta_2 - 3\beta_1)\Gamma \quad \bar{b}_1 = -\bar{a} \quad \bar{b}_2 = -(2\beta_2 - 3\beta_1 - 6)/\Gamma. \tag{16}$$

Since we are working with raw moments, we make the transformation $\bar{x} = x - \mu'_1$ in equation (16), obtaining

$$\frac{1}{f(\beta, x)}\frac{df}{dx} = \frac{x + \bar{a} - \mu'_1}{(\bar{b}_0 - \bar{b}_1\mu'_1 + \bar{b}_2\mu'^2_1) + (\bar{b}_1 - 2\bar{b}_2\mu'_1)x + \bar{b}_2x^2} \tag{17}$$

$$= \frac{x + a}{b_0 + b_1x + b_2x^2} \tag{18}$$

if we set

$$a = \bar{a} - \mu'_1 \quad b_0 = \bar{b}_0 - \bar{b}_1\mu'_1 + \bar{b}_2\mu'^2_1 \quad b_1 = \bar{b}_1 - 2\bar{b}_2\mu'_1 \quad b_2 = \bar{b}_2 \tag{19}$$

Equations (19) and (16) together relate (a, b_0, b_1, b_2) to the more familiar parametrisation $(\mu'_1, \sigma^2, \beta_1, \beta_2)$. The virtue of the (a, b_0, b_1, b_2) form is that it yields a recurrence relation for the raw moments of a Pearson curve (Elderton/Johnson, pp 37-38):

$$[(n+2)b_2 + 1]\mu'_{n+1} = -(a + (n+1)b_1)\mu'_n - nb_0\mu'_{n-1} \tag{20}$$

which we will use for $n+1 = 2, \ldots, 8$.

§§2.2 Higher moments of the Pearson type IV family - an alternative method (not implemented)

The higher (theoretical) moments of a Pearson family can be computed via recurrence relations such as those given on pp 37-38 of Elderton/Johnson (1969). These relations, as we have seen require determination of a, b_0, b_1, b_2 in terms of the input parameters $\mu, \sigma^2, \beta_1, \beta_2$. As a check on these calculations, one can use the simpler recurrence relation (for the Type IV family) given on p 64 of Elderton/Johnson, namely

$$\tilde{\mu}_n = \frac{a}{r-n+1}\{(n-1)a\tilde{\mu}_{n-2} - \nu\tilde{\mu}_{n-1}\} \tag{21}$$

which is expressed in terms of the parameters a, ν, r that are obtained directly from $\sigma^2, \beta_1, \beta_2$. The catch is that the moments $\tilde{\mu}_n$ are calculated relative to an origin located $\nu a/r$ to the right of the mean. Thus, to obtain the raw moments μ_n' (about the original origin), a transformation is necessary. If $\tilde{x}$ has origin $\nu a/r$ to the right of the mean, then it is related to the original coordinate x by $\tilde{x} = x - \mu - \frac{\nu a}{r} = x - z$ say. We are interested in

$$\begin{aligned}
\sigma_{kk'} = \mathrm{Cov}(X^k, X^{k'}) &= \mathrm{Cov}((\tilde{X}+z)^k, (\tilde{X}+z)^{k'}) \\
&= \sum_{j=1}^{k}\sum_{j'=1}^{k'} \binom{k}{j} z^{k-j} \mathrm{Cov}(\tilde{X}^j, \tilde{X}^{j'}) \binom{k'}{j'} z^{k'-j'} \\
&= \sum_j \sum_{j'} a_{kj}\tilde{\sigma}_{jj'}a_{k'j'}.
\end{aligned}$$

Thus if we set $\Sigma = (\sigma_{kk'})_{4\times4}, \tilde{\Sigma} = (\tilde{\sigma}_{jj'})_{4\times4}, A = (a_{kj})_{4\times4}$ (where $a_{kj} = \binom{k}{j} z^{k-j} I(k \geq j)$), then we can write the above in matrix form as

$$\Sigma = A\tilde{\Sigma}A^t.$$

Here $\tilde{\Sigma}$ is known easily, since $\tilde{\sigma}_{jj'} = \tilde{\mu}_{j+j'} - \tilde{\mu}_j\tilde{\mu}_{j'}$, which may be read off from (21) above. Matrix multiplication routines can then be used to obtain Σ.

§§2.3 Inputs to numerical integration routines - Type IV Pearson curves

Here we suppose that X has the type IV Pearson density (in the form given by Elderton/Johnson (1969, p 58))

$$f(\tilde{x}) = c\left(1 + \frac{\tilde{x}^2}{a^2}\right)^{-\frac{r+2}{2}} \exp(-\nu \tan^{-1}(\tilde{x}/a)) \qquad -\infty < \tilde{x} < \infty \tag{22}$$

where

$$(23)\qquad \begin{aligned} r &= \frac{6(\beta_2-\beta_1-1)}{2\beta_2-3\beta_1-6} \\ \nu &= \frac{r(r-2)\sqrt{\beta_1}}{\Delta} \\ a &= \frac{\sigma\Delta}{4} \\ \Delta &= \{16(r-1)-\beta_1(r-2)^2\}^{1/2} \end{aligned}$$

Note that the origin is $\nu a/r$ after the mean μ_1 and that the argument $\tilde{x}$ is referred to this origin. Note also that $\sigma^2, \beta_1, \beta_2$ are the usual variance (squared) skewness and kurtosis, given in terms of central moments μ_k by μ_2, μ_3^2/μ_2^3 and μ_4/μ_2^2 respectively. The normalization constant c depends on $\sigma^2, \beta_1, \beta_2$, and will be calculated in the numerical integration routine.

As demonstrated on p 64 of Elderton/Johnson (1969), cumulative probabilities for the density $f(\beta,\tilde{x})$ may be most easily computed after the changes of variables $\tan\theta = \tilde{x}/a$ and $\pi/2 = \theta+\phi$, leading to the equivalent expressions

$$(24)\qquad \begin{aligned} \tilde{F}(\beta,\tilde{x}_0) &= \int_{-\infty}^{\tilde{x}_0} f(\beta,\tilde{x})d\tilde{x} \\ &= \int_{\pi/2}^{\theta_0} \cos^r(\theta)e^{-\nu\theta}d\theta \Big/ \int_{-\pi/2}^{\pi/2}\cos^r\theta e^{-\nu\theta}d\theta \quad , \quad \theta_0 = \tan^{-1}(\tilde{x}_0/a) \\ &= \int_{\phi_0}^{\pi} \sin^r(\phi)e^{\nu(\phi-\pi/2)}d\phi \Big/ \int_0^{\pi}\sin^r\phi e^{\nu(\phi-\frac{\pi}{2})}d\phi \quad , \quad \phi_0 = \frac{\pi}{2}-\tan^{-1}(\tilde{x}_0/a) \\ &= I(\phi_0,r,\nu)/I(0,r,\nu). \end{aligned}$$

If the mean is allowed to be arbitrary (as we do here to permit flexibility), then an initial change of co-ordinates is required. Given parameters $\beta = (\mu,\sigma^2,\beta_1,\beta_2)$, we obtain

$$(25)\qquad \begin{aligned} F(\beta,x) &= P_B(X\le x) \\ &= P(X-(\mu+\frac{\nu a}{r}) \le x-(\mu+\frac{\nu a}{r})) \\ &= \tilde{F}(\beta,\tilde{x}) \end{aligned}$$

where $\tilde{x} = x-(\mu+\nu a/r)$. Differentiating, we get

$$(26)\qquad f(\beta,x) = \tilde{f}(\beta,\tilde{x}).$$

§3. Specific details of the program

As described in the theoretical development of the previous section, the basic goal of the program is the calculation of

$$(27) \qquad \mathrm{Var}_\beta(x_\epsilon(\hat{\beta})) \doteq \frac{1}{f^2(\beta, x_\epsilon(\beta))}[D_1F]^t[D\mathcal{G}][DH]\mathrm{Var}_\beta(\hat{\mu}')[DH]^t[D\mathcal{G}]^t[D_1F]$$

(combine formulas (7) and (12)).

The labels A-D correspond to labels in the program comments.

A: a) The program requests input parameters:

$\beta = (\mu, \sigma^2, \beta_1, \beta_2)$	- parameters describing the Pearson-type IV distribution which is assumed to generate the data. (There is a check that they are in the type IV region.)
nsamp	- sample size available for estimating $\mu_1', \mu_2', \mu_3', \mu_4'$
ϵ=upper percentile	- entering say, $\epsilon = .005$ would ask for the upper 1/2% point of the specified type IV curve.

b) The zero-finding subroutine is used to find the ϵ^{th} quantile x_0, solving

$$F_\beta(x_0) = \epsilon.$$

B: This section is largely self-explanatory, assembling components for the basic calculation. Computation of $D\mathcal{G}$ and DH is straightforward, but some care with co-ordinate systems is needed in evaluation of $\mu_k', k = 1, \ldots, 8$; the raw moments of the theoretical type IV distribution. The μ_k' are found by a recurrence relation (20) in the original co-ordinate system. To express the coefficients a, b_0, b_1, b_2 in (20) in terms of the input data $\mu, \sigma^2, \beta_1, \beta_2$, a detour is made through a co-ordinate system referred to the mean μ (see formulas (15)-(19)).

C. This section performs the numerical differentiation (rather crudely!). Here is the principle used: imagine a real valued function $g(z_1, \ldots, z_p)$. The i^{th} partial derivative at $z^0 = (z_1^0, \ldots, z_p^0)$ is estimated as

$$\frac{\partial g}{\partial z_i}(z^0) \doteq \frac{g(z^0 + \epsilon_r e_i) - g(z^0 - \epsilon_l e_i)}{\epsilon_l + \epsilon_r}$$

Here $e_i = (0, \ldots, 1, \ldots, 0)$ is the i^{th} unit co-ordinate vector, and $\epsilon_l, \epsilon_r \geq 0$ are non-negative, not necessarily equal, and not both zero.

Checks are performed that the arguments $z^0 + \epsilon_r e_i$, $z^0 - \epsilon_l e_i$ lie within the allowable Pearson type IV range, if they do not, then they are adjusted so that they do.

The size of the steps ϵ_l, ϵ_r are governed (in the absence of the restrictions mentioned above) by a small parameter δ, and the size of the argument. Thus, in approximating the i^{th} partial derivative at z^0

$$\epsilon_r = \max(\delta, \delta z_i^0).$$

D. Final evaluation.

The matrix multiplications are done using IMSL routines for computing AB, AB^t, A^tB. (documentation attached). The final step is to compute the denominator in (27): the population density at the quantile, and this is done using formula (22). Note that the normalizing constant

$$c = \frac{1}{a} I(0, r, \nu)$$

(see notation of subsection 2.3).

OUTRANGE (β_1, β_2)

This subroutine (which is invoked both at initial input of parameters, and when parameters are modified during the computation of numerical derivatives) checks whether β_1, β_2 lie in the Type IV region (or the type III boundary corresponding to the t-family, when $\beta_1 = 0$). Thus the routine checks if

(i) $\beta_1 \geq 0$, $\beta_2 \leq 3$

(ii) $0 \leq k < 1$, where $k = \frac{\beta_1(\beta_2+3)^2}{4(4\beta_2-3\beta_1)(2\beta_2-3\beta_1-6)}$

(A picture of the region defined by (i) and (ii) is given in Volume 2 of the *Biometrika Tables for Statisticians*).

ZERO $(\mu, \sigma^2, \beta_1, \beta_2, \epsilon, x_0)$

This subroutine, for given Pearson curve parameters $\mu, \sigma^2, \beta_1, \beta_2$, and given tail probability ϵ, solves for the corresponding quantile x_0 such that

$$F_\beta(x_0) = P_{\mu,\sigma^2,\beta_1,\beta_2}(X \leq x_0) = \epsilon.$$

It uses the IMSL routine ZBRENT, which uses (I think) a combination of bisection, linear and inverse quadratic interpolation to find the zero of an equation - in this case, the equation $F_\beta(x) - \epsilon = 0$.

This equation clearly has a unique root, and ZBRENT requires two initial values LEFT and RIGHT, which must be of opposite sign. Somewhat arbitrarily, the present code takes

$$\text{LEFT} = \mu$$

$$\text{RIGHT} = \mu + 12\sigma$$

on the assumption that ϵ will be chosen so that we are working in the right tail of the distribution, and that 12 standard deviations to the right of the mean should leave a smaller remaining tail area than any conceivably realistic choice of ϵ.

Of course, ZBRENT repeatedly calls and evaluates the function $P4DF$ - the cumulative distribution function of a Pearson type IV curve.

$P4DF(\mu, \sigma^2, \beta_1, \beta_2, x)$

This function computes the c.d.f. of a random variable X following a Pearson type IV curve described by parameters $\beta = (\mu, \sigma^2, \beta_1, \beta_2)$ and returns the value $F_\beta(x) = P_\beta(X \leq x)$.

Subsection 2.3: "Inputs to numerical integration routines : type IV Pearson curves" describes the form in which the type IV density is used in the numerical integration. The actual calculation is done by the IMSL routine DMLIN (see documentation attached) which employs Gaussian integration separately on the numerator and denominator in the expression (24)

$$\tilde{F}(\beta, \tilde{x}^0) = I(\phi_0, r, \nu)/I(0, r, \nu).$$

References

Rao, C.R. (1973) Linear Statistical Inference and Its Applications, 2nd edn., Wiley, New York.

Elderton, W.P., and Johnson, N.L. (1969) Systems of Frequency Curves. Cambridge University Press.

APPENDIX

```
      program quantile

      real*8  mp(8)
      real*8  mu(4)
      real*8  mm,sig2,beta1,beta2,rhs,x0
      real*8  dh(4,4)
      real*8  dg(4,4)
      real*8  r,nu,a,del
      real*8  aa,b0,b1,b2
      real*8  num,denom
      real*8  cov(4,4)
      real*8  gam,abar,b0bar,b1bar,b2bar
      real*8  nsamp

c     declarations for matrix multiplication, numerical derivatives
c     and calculation of density

      real*8  t1(4,4), t2(1,4), t3(1,4), t4(1,1)
      real*8  delta, epsl,epsr, df(4,1),p4df,small

      integer l,m,n,ia,ib,ic,ier
      integer maxfcn,nn,iier
      real*8  dmlin,low(1),up(1),aerr,rerr,integ
      real*8  pi,piby2,x,temp,dens
      external f
      logical outrange,long
      common r,nu,piby2
      character letter

      data    dh/16*0.0/
      data    dg/16*0.0/

c     these computations would be needed were we to
c     read in raw moments muprime(1:4) >
c
c     <compute first four centered moments>
c
c     mu(1) = mp(1)
c     mu(2) = mp(2) - mp(1)**2
c     mu(3) = mp(3) - 3*mp(2)*mp(1) + 2*mp(1)**3
c     mu(4) = mp(4) - 4*mp(3)*mp(1) + 6*mp(2)*mp(1)**2 - 3*mp(1)**4
c
c     <compute mean variance beta1,beta2
c
c     mm = mu(1)
c     sig2 = mu(2)
c     beta1 = mu(3)**2 / mu(2)**3
c     beta2 = mu(4) / mu(2)**2

      print *, "long output? "
      read * , letter
      long = letter .eq. 'y'

      write (6,13) "mean","var","beta1","beta2","nsamp","eps","x0",
     +                                  "dens","s.d."
      do 10, iter = 1,10000
```

```
c       ***************
c       SECT A. : read in mean,variance,beta1,beta2, x0,upper %tile

        if (long) print *, "input mean,var,beta1,beta2,sampsize,upper %tile"
        read (5,*,end =15 )  mm,sig2,beta1,beta2,nsamp,rhs
        if (long) print *, mm,sig2,beta1,beta2,nsamp,rhs
        if (outrange(beta1,beta2)) print *,"warning: parms outside typeIV "

c       now find quantile x0 yielding tail probability rhs

        call zero(mm,sig2,beta1,beta2,rhs,x0)

c       ***********************
c       SECT. B. : calculate centered moments thru order four
c       using formula (8)

        mu(1) = mm
        mu(2) = sig2
        mu(3) = sqrt( beta1 * mu(2)**3 )
        mu(4) = beta2 * mu(2)**2
        if (long)  print *, "mu = "
        if (long) write (6,44) (mu(i),i=1,4)
 44     format (4F14.6)

c       calculate raw moments thru order four using formula (11)

        mp(1) = mu(1)
        mp(2) = mu(2) + mp(1)**2
        mp(3) = mu(3) + 3*mp(1)*mp(2) - 2*mp(1)**3
        mp(4) = mu(4) + 4*mp(3)*mp(1) - 6*mp(2)*mp(1)**2 + 3*mp(1)**4

c       compute entries in Jacobian matrix for H using formula (13)

        dh(1,1) = 1.
        dh(2,2) = 1.
        dh(3,3) = 1.
        dh(4,4) = 1.
        dh(2,1) = - 2*mp(1)
        dh(3,1) = 6*mp(1)**2 - 3*mp(2)
        dh(3,2) = - 3*mp(1)
        dh(4,1) = (-4*mp(3)) + 12*mp(2)*mp(1) - 12*mp(1)**3
        dh(4,2) = 6*mp(1)**2
        dh(4,3) = -4*mp(1)

        if (long) print *, "dh ="
        if (long)  write (6,140) ((dh(i,j),j=1,4),i=1,4)
 140    format (/(4F14.6) )

c       compute entries in Jacobian matrix for  G  using formula (10)

        dg(1,1) = 1.
        dg(2,2) = 1.
        dg(3,2) = (-3*mu(3)**2 ) / (mu(2)**4 )
        dg(3,3) = ( 2*mu(3) ) /  (mu(2)**2)
        dg(4,2) = ( -2*mu(4) ) / ( mu(2)**3 )
```

```
        dg(4,4) = 1./ (mu(2)**2 )
        if (long) print *, "dg ="
        if (long) write (6,140) ((dg(i,j),j=1,4),i=1,4)

c       compute r,nu,a,del - parameters in type IV density using formula (A.2)

        r = (6.*(beta2 - beta1 - 1.)) / ( 2.*beta2 - 3.*beta1 - 6.)
        del = sqrt( 16.*(r-1.) - beta1*(r-2.)**2 )
        nu = ( (-r)*(r-2.)*sqrt(beta1) )/del
        a = sqrt(sig2) * del / 4.

c       check to see that 8th moment exists (it often doesn't)

        if ( r .le. 7) then
                print * ," 8th moment infinite : abort "
                stop
        end if

c       set to work on higher moments via the recurrence relations
c       begin in scale centered on mean - see formulas (16)

        gam = 2.*(5*beta2 - 6*beta1 - 9.)
        abar = sqrt(sig2 * beta1) * (beta2 + 3.) /gam
        b0bar =(- sig2)*( 4*beta2 - 3*beta1 ) / gam
        b1bar = - abar
        b2bar = - ( 2*beta2 - 3*beta1 - 6.) / gam

c       now convert to absolute origin using formulas (19)

        aa = abar - mp(1)
        b0 = b0bar - b1bar*mp(1) + b2bar*mp(1)**2
        b1 = b1bar - 2*b2bar*mp(1)
        b2 = b2bar

c       now set up recurrence relation using formula (20) :
c       note  mp(1) through mp(4) were calculated earlier

        do 20, i = 5,8
                num = ( aa + i*b1 )*mp(i-1)  +  (i-1)*b0*mp(i-2)
                denom = (i+1)*b2 + 1.
                mp(i) = (-num)/denom
20      continue
        if (long) print *, "mp = "
        if (long) write (6,80) (mp(i),i=1,8)
80      format (4f14.6)

c       set up covariance matrix of raw sampling moments (cf. formula (14) )

        do 30, i = 1,4
                do 40, j = 1,4
                        cov(i,j) = ( mp(i+j) - mp(i)*mp(j) )/nsamp
40              continue
30      continue
        if (long) print *, "cov ="
        if (long) write (6,140) ((cov(i,j),j=1,4),i=1,4)
```

```
c       *************************
c       SECT. C. : It's time to compute df : numerical derivatives of
c       Pearson cdf wrt mean,var,beta1,beta2


        delta = .1
        epsr = max( delta, mm * delta )
        epsl = epsr
        df(1,1) =    ( p4df(mm+epsr,sig2,beta1,beta2,x0)  -
     +                 p4df(mm-epsl,sig2,beta1,beta2,x0)  )/ (epsl+epsr)

        epsr = max( delta, sig2 * delta )
        epsl = epsr
        if (epsl .gt. .25*sig2 )  epsl = .25*sig2
        df(2,1) =    ( p4df(mm,sig2+epsr,beta1,beta2,x0)  -
     +                 p4df(mm,sig2-epsl,beta1,beta2,x0)  )/ (epsl+epsr)

        small = 1.d-7
        do 1001 , i = 1,8
             epsr = max( delta, beta1 * delta )
             epsl = epsr
             if ( outrange(beta1+epsr,beta2) )  epsr = 0.
             if ( outrange(beta1+epsl,beta2) )  epsl = 0.
             if ( (epsl+epsr) .gt. small ) goto 1002
             if ( i .eq. 8 ) then
                print *, " unfixable epsl+epsr = 0 in del(beta1)  : aborted "
                stop
             end if
             delta = delta/2.
 1001           continue

 1002   df(3,1) =    ( p4df(mm,sig2,beta1+epsr,beta2,x0)  -
     +                 p4df(mm,sig2,beta1-epsl,beta2,x0)  )/ (epsl+epsr)

        delta = .1
        do 1011 , i = 1,8
             epsr = max( delta, beta2 * delta )
             epsl = epsr
             if ( outrange(beta1,beta2+epsr) )  epsr = 0.
             if ( outrange(beta1,beta2-epsl) )  epsl = 0.
             if ( (epsl+epsr) .gt. small ) goto 1012
             if ( i .eq. 8 ) then
                print *, " unfixable epsl+epsr = 0 in del(beta2)  : aborted "
                stop
             end if
             delta = delta/2.
 1011           continue

 1012   df(4,1) =    ( p4df(mm,sig2,beta1,beta2+epsr,x0)  -
     +                 p4df(mm,sig2,beta1,beta2-epsl,x0)  )/ (epsl+epsr)

        if (long) print *, " df = "
        if (long) write (6,120) ((df(i,j),i=1,4),j=1,1)
 120    format (/(4f14.6) )

c       *********************
```

```
c       SECT D. · matrix multiplications:
c       t1 = dg * dh
c       t2 = t(df) * t1
c       t3 = t2 * cov
c       t4 = t3 * t(t2)


        l = 4
        m = 4
        n = 4
        ia = 4
        ib = 4
        ic = 4
        call vmulff(dg,dh,l,m,n,ia,ib,t1,ic,ier)
        if ( ier .ne. 0 ) print * , "ier = ", ier

        l = 4
        m = 1
        n = 4
        ia = 4
        ib = 4
        ic = 1
        call vmulfm(df,t1,l,m,n,ia,ib,t2,ic,ier)
        if ( ier .ne. 0 ) print * , "ier = ", ier

        l = 1
        m = 4
        n = 4
        ia = 1
        ib = 4
        ic = 1
        call vmulff(t2,cov,l,m,n,ia,ib,t3,ic,ier)
        if ( ier .ne. 0 ) print * , "ier = ", ier

        l = 1
        m = 4
        n = 1
        ia = 1
        ib = 1
        ic = 1
        call vmulfp(t3,t2,l,m,n,ia,ib,t4,ic,ier)
        if ( ier .ne. 0 ) print * , "ier = ", ier

        if (long) print *, " t1 = "
        if (long) write (6,140) ((t1(i,j),j=1,4),i=1,4)

        if (long) print *, " t2 = "
        if (long) write (6,120) (t2(1,j),j=1,4)

        if (long) print *, " t3 = "
        if (long) write (6,120) (t3(1,j),j=1,4)

        if (long) print *, " t4 = "
        if (long) write (6,140) t4(1,1)

c       compute density at x0, obtaining dens , using formula (A.1)
```

```
      pi = 4. * atan(1.)
      piby2 = pi/2.
      x = x0 - ( mm + (nu*a/r) )
      temp = ((1. + (x/a)**2) ** (-(r + 2.)/2.)) *exp(-nu* atan(x/a))

c     now compute normalising constant in density

      low(1) = 0.
      up(1) = pi
      nn = 1
      rerr = 1.d-7
      aerr = 0.0
      maxfcn = 10000
      integ = dmlin(f,low,up,nn,maxfcn,aerr,rerr,iier)

      dens = temp / (a * integ)
      if (long) print *, "dens =  ", dens

      result = t4(1,1)/ (dens**2)
      if (long) then
                  print *,"x0= ",x0,", and has asy. s.d.= ",sqrt(result)
              else
                  write (6,12) mm,sig2,beta1,beta2,nsamp,rhs,x0,
     +                                          dens,sqrt(result)
      end if
 12   format (1H ,4F7.2,1H  ,5G10.3e1)
 13   format (4A7,1H ,5A9 )

 10   continue

 15   print * , "    "
      end
```

```
      real*8 function f(n,x)
      real*8  x(n),r,nu,piby2
      common r,nu,piby2

      f =   sin(x(1))**r * exp( (x(1)-piby2)*nu )

      return
      end


      logical function outrange(betal,beta2)

c     checks to see if the skewness and kurtosis lie within the range
c     allowed for the Pearson type IV system: check is based on the kappa
c     criterion given in Elderton/Johnson p 45

      real*8 betal,beta2,num,denom,ratio

      num = betal*(beta2 + 3.) ** 2
      denom = 4.*( 4.*beta2 - 3.*betal) * ( 2.*beta2 - 3.*betal - 6.)
      ratio = num/denom
      outrange = betal .lt. 0.  .or.  ratio .ge. 1.  .or.  ratio .lt. 0.
     +               .or. beta2 .lt. 3.0
      return
      end
```

```
      real*8 function p4df(mm,sig2,betal,beta2,x0)

      real*8  mm,sig2,betal,beta2,x0
      real*8 r,nu,a,del
      integer maxfcn,n,ier
      real*8  dmlin,low(1),up(1),aerr,rerr
      real*8  pi,piby2,phi,x
      real*8  num,denom
      common r,nu,piby2
      external f

      n = 1
      rerr = 1.d-7
      maxfcn = 10000

c     compute r,nu,a,del - parameters in type IV density

      r = (6.*(beta2 - betal - 1.)) / ( 2.*beta2 - 3.*betal - 6.)
      del = sqrt( 16.*(r-1.) - betal*(r-2.)**2 )
      nu = ( (-r)*(r-2.)*sqrt(betal) )/del
      a = sqrt(sig2) * del / 4.

c     print *, "r = ",r,"nu= ",nu,"a= ",a

c     transform origin to canonical form , and compute lower limit phi

      pi = 4. * atan(1.)
      piby2 = pi/2.
      x = x0 - ( mm + (nu*a/r) )
      phi = piby2 - atan(x/a)

      up(1) = phi
      low(1) = 0.

      num = dmlin(f,low,up,n,maxfcn,aerr,rerr,ier)
      if  ( ier .ne. 0 )  then
            print *, " ier = " , ier
      end if

      up(1) = pi
      denom = dmlin(f,low,up,n,maxfcn,aerr,rerr,ier)
      if  ( ier .ne. 0 )  then
            print *, " ier = " , ier
      end if
      p4df= num/denom
      return
      end
```

```
      subroutine zero(mm,sig2,betal,beta2,rhs,x0)

      integer nsig,maxfn,ieer
      real*8  eps,left,right,funct
      real*8  mm,sig2,betal,beta2,rhs,x0
      real*8  tmm,tsig2,tbetal,tbeta2,trhs
      external funct
      common /brent/ tmm,tsig2,tbetal,tbeta2,trhs

      tmm=mm
      tsig2=sig2
      tbetal=betal
      tbeta2=beta2
      trhs=rhs

      eps = 0.0
      nsig = 5
      left = mm
      right = mm + 12.*sqrt(sig2)
      maxfn = 200

      call zbrent( funct,eps,nsig,left,right,maxfn,ieer )

      x0 = right
c     print *, " zero is at ", x0

      end


      real*8 function funct(y)

      real*8 tmm,tsig2,tbetal,tbeta2,trhs,y,p4df
      common /brent/ tmm,tsig2,tbetal,tbeta2,trhs

      funct = p4df(tmm,tsig2,tbetal,tbeta2,y) - trhs

      return
      end
```

AMENDED

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 381	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Program For Estimating Uncertainties In Quantile Estimates Derived From Empirical Pearson Fits		5. TYPE OF REPORT & PERIOD COVERED TECHNICAL REPORT 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Iain Johnstone		8. CONTRACT OR GRANT NUMBER(s) N00014-86-K-0156
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Statistics Stanford University Stanford, CA 94305		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR-042-267
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Statistics & Probability Program Code 1111		12. REPORT DATE October 14, 1986 13. NUMBER OF PAGES 26
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Partial support supplied by National Science Foundation under Reference No. DMS-8451750.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Pearson curve, delta method, estimated quantile, method of moments, asymptotic variance

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Many authors have warned of the hazards involved in fitting Pearson curves on the basis of empirically determined moments. For the problem of estimating a percentage point of a Pearson curve on the basis of empirical moment estimates, the "delta method" easily yields an approximation to the variability of the estimated quantile. This report describes a crude program for assessing quantitatively the size of these hazards in the case of Pearson Type IV curves (an important family including the student t family as a boundary case).

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

END

12-86

DTIC